Claim Amendments:

Claim 1. (currently amended) A method comprising:

receiving a request on a platform having a battery mode to perform a write operation to a file system device;

determining whether the file system device is activated or inactivated;

if the file system device is determined to be activated, accessing the file system device to perform the requested write operation; and

if the file system device is determined to be inactivated, buffering the write operation to physical memory,

thereby extending battery life of the platform while minimizing adverse effects on performance and/or functionality of the platform.

Claim 2. (currently amended) A method comprising:

receiving a request in a host processor to perform a write operation to a device communicatively coupled to the host processor;

determining whether the device is activated or inactivated;

if the device is determined to be activated, accessing the device to perform the requested write operation;

if the device is determined to be inactivated, buffering the write operation to physical memory coupled to the host processor, by an intermediate file system driver (FSD) executing on the host processor; and

determining whether the device is operating in a limited power state prior to determining whether the device is activated or inactivated.

Claim 3. (previously amended) The method of claim 2 wherein the device comprises a disk drive, a non-volatile memory component, or a network access device.

Claim 4. (previously amended) The method of claim 2 wherein determining whether the device is activated or inactivated comprises determining whether the device is powered-up or powered-down, respectively.

Claim 5. (currently amended) The method of claim 2 wherein receiving a request to perform a write operation comprises using ~~an~~ the intermediate file system driver to intercept a request bound for a file system driver.

Claim 6. (previously amended) The method of claim 2 further comprising:

writing one or more buffered write operations to the device upon an occurrence of a predetermined condition after activating the device if the device is determined to be inactivated.

Claim 7. (currently amended) A method comprising:

receiving a request in a host processor to perform a write operation to a device communicatively coupled to the host processor;

determining whether the device is activated or inactivated;

if the device is determined to be activated, accessing the device to perform the requested write operation; and

if the device is determined to be inactivated, buffering the write operation to physical memory coupled to the host processor, by an intermediate file system driver (FSD) executing on the host processor; and

writing one or more buffered write operations to the device upon an occurrence of a predetermined condition after activating the device if the device is determined to be inactivated,

wherein the predetermined condition is identified by the intermediate FSD,

wherein the buffered write operations are stored in physical memory, and

wherein the predetermined condition comprises one or more of the following: detecting that a memory write buffer is full, detecting that a predetermined amount of time has lapsed, detecting that a predetermined volume of data has been buffered, detecting that battery power is at a threshold level, detecting that a computer system with which the device is associated is being

turned off or put in a standby state, and detecting an explicit request that the write buffer contents be committed to non-volatile storage.

Claim 8. (previously amended) The method of claim 7 further comprising receiving user input requesting that the buffered write operations be committed to non-volatile storage; and

detecting an input request that the write buffer contents be committed to non-volatile storage.

Claim 9. (currently amended)

A method comprising:

receiving a request in a host processor to perform a write operation to a file system device communicatively coupled to the host processor;

determining whether the file system device is activated or inactivated;

if the file system device is determined to be activated, accessing the file system device to perform the requested write operation;

if the file system device is determined to be inactivated, buffering the write operation to physical memory coupled to the host processor, by an intermediate file system driver (FSD) executing on the host processor; and

determining whether the requested write operation corresponds to an entity registered with the intermediate FSD to participate in the method of controlling device write operations, the controlling being performed by the intermediate FSD.

Claim 10. (currently amended)

A method comprising:

receiving a request in a host processor to perform a write operation to a file system device communicatively coupled to the host processor;

determining whether the file system device is activated or inactivated;

if the file system device is determined to be activated, accessing the file system device to perform the requested write operation; and

if the file system device is determined to be inactivated, buffering the write operation to physical memory coupled to the host processor, by an intermediate file system driver (FSD) executing on the host processor, wherein buffering the write operation to physical memory comprises deleting from physical memory a prior buffered write operation request that seeks to modify a same storage location on the device as the write operation to be buffered.

Claim 11. (currently amended) A method comprising:

receiving a request from a process executing on a host processor to read a portion of a file from a device communicatively coupled to the host processor;

determining whether a limited power condition exists;

if a limited power condition is determined not to exist, accessing the device to read the requested file portion into memory; and

if a limited power condition is determined to exist, accessing the device to read a superset of the requested file portion into memory coupled to the host processor, wherein the superset of the requested file portion is logically related to the requested portion, and wherein the superset is selectively determined by an intermediate file system driver (FSD) executing on the host processor.

Claim 12. (original) The method of claim 11 wherein reading a superset of the requested file portion into memory comprises reading the entire file into memory.

Claim 13. (original) The method of claim 11 wherein reading a superset of the requested file portion into memory comprises reading a subset of the entire file into memory.

Claim 14. (original) The method of claim 13 further comprising identifying the subset of the entire file to be read into memory.

Claim 15. (currently amended) The method of claim 14 wherein identifying the subset of the entire file to be read into memory is based on one or more temporal file access trends.

Claim 16. (original) The method of claim 11 further comprising returning the requested file portion to the requesting process.

Claim 17. (original) The method of claim 11 wherein, if a limited power condition exists, the requested file portion is read from the device and returned to the requesting process before a remainder of the superset is read into memory.

Claim 18. (original) The method of claim 11 wherein, if a superset of the requested file portion is read into memory, further comprising accessing the superset read into memory to fulfill a subsequent request from the process for a portion of the file.

Claim 19. (original) The method of claim 11 further comprising, if a superset of the requested file portion is read into memory, deactivating the device.

Claim 20. (original) The method of claim 11 wherein the device comprises a disk drive or a network access device.

Claim 21. (original) The method of claim 11 wherein reading the superset of the requested file portion into memory comprises translating the received read request for the file portion into a plurality of read requests that collectively cause the superset to be read from the device.

Claim 22. (currently amended) The method of claim 11 further comprising determining whether the requested read operation corresponds to a file type registered with the intermediate file system driver to participate in the method of controlling device read operations.

Claim 23. (currently amended) The method of claim ~~11~~ 22 wherein each of a plurality of file types has an associated priority and wherein the method further comprising selectively storing the superset of the requested file portion into memory based on its relative priority.

Claim 24. (currently amended) A system comprising:

at least one non-volatile storage device;

a physical memory ; and

a processor communicatively coupled to the physical memory and the at least one non-volatile storage device, the processor to execute instructions to perform operations comprising:

detecting that a time-out period is to expire for deactivating a non-volatile storage device that provides access to data;

determining whether a limited power condition exists; and

if a limited power condition is determined to exist and a time-out period is about to expire, writing one or more buffered write operations from physical memory to the non-volatile storage device before the non-volatile storage device is deactivated, wherein the writing is in response to an intermediate file system driver detecting that a time-out has occurred..

Claim 25. (previously amended) The system of claim 24 wherein the non-volatile storage device that provides access to data comprises a disk drive or a network access device.

Claim 26. (previously amended) The system of claim 24 wherein determining whether a limited power condition exists comprises determining whether a system associated with the non-volatile storage device is operating under battery power.

Claim 27. (currently amended) A machine-accessible medium embodying instructions for causing a machine to perform operations comprising:

determining a power state of a non-volatile storage device;

selectively buffering a file system write request, in physical memory, the write request relating to the non-volatile storage device based on the determined power state of the non-volatile storage device; and

determining whether the device is operating in a limited power state prior to determining whether the device is activated or inactivated,

wherein the buffering prevents unnecessary activation of the non-volatile storage device.

Claim 28. (previously amended) The medium of claim 27 wherein determining a power state of a non-volatile storage device comprises determining whether the non-volatile storage device is operating under battery power.

Claim 29. (currently amended) The medium of claim 28 further comprising instructions for writing one or more buffered write ~~operations~~ requests to the non-volatile storage device upon an occurrence of a detected predetermined condition,

wherein the predetermined condition comprises at least one condition selected from a group of conditions consisting of (i) detecting that a write buffer has become full, (ii) detecting that a certain amount of time has passed, (iii) detecting that battery power is approaching a specified threshold level, (iv) detecting that the machine is being turned off, (v) detecting that the machine is to be put in a standby state, and (vi) detecting that one of a user, a process and an operating system has explicitly requested that the write buffer contents be committed to non-volatile storage by the device.

Claim 30. (currently amended) The medium of claim 29 further comprising instructions for causing a machine to deactivate the non-volatile storage device after writing the one or more buffered write ~~operations~~ requests, the machine to continue operations after deactivating the non-volatile storage device.

Claim 31. (previously amended) A machine-accessible medium embodying instructions for causing a machine to perform operations comprising:

determining a power state of a device; and

based on the determined power state of the device and in response to a file system request to read a portion of the file from the device, selectively reading a superset of the requested file portion from the device into physical memory, wherein the superset of the requested file portion is logically related to the requested portion.

Claim 32. (original) The medium of claim 31 wherein selectively reading a superset of the requested file portion comprises reading the entire file into physical memory.

Claim 33. (original) The medium of claim 31 wherein determining a power state of a device comprises determining whether the device is operating under battery power.

Claim 34. (original) The medium of claim 31 wherein selectively reading a superset of the requested file portion from the device into physical memory comprises translating the file system request to read a portion of the file portion into a plurality of read requests that collectively cause the superset to be read from the device.

Claim 35. (original) The medium of claim 31 wherein the requested file portion is read from the device and returned to a requesting process before a remainder of the superset is read into physical memory.

Claim 36. (original) The medium of claim 31 further comprising accessing the superset read into physical memory to fulfill a subsequent file system request to read a portion of the file.

Claim 37. (previously amended) The medium of claim 31 further comprising instructions for causing a machine to deactivate the device after reading the superset of the requested file portion.

Claim 38. (canceled)

Claim 39. (currently amended) A system, comprising:

a processor communicatively coupled to physical memory;

a non-volatile storage device communicatively coupled to the processor, wherein access to the non-volatile storage device is controlled by a file system driver responsive to file system requests; and

an intermediate file system driver to receive user customized parameters and to receive file system requests, the intermediate file system driver to ~~provide~~ determine read/write policy ~~to~~ for controlling access to the file system driver based on the user customized parameters, wherein file system requests are to be intercepted by the intermediate file system driver, wherein the file system driver accesses the non-volatile storage device in accordance with the read/write policy, and wherein the read/write policy is to minimize at least one of (a) unnecessary device access operations and (b) unnecessary device activation-deactivation operations,

wherein the intermediate file system driver intercepts a file system write request and selectively buffers the write request to physical memory until a predetermined condition is detected, wherein responsive to the predetermined condition, the intermediate file system driver initiates performance of the write request of the buffered write request.

Claim 40. (canceled)

Claim 41. (currently amended) The system as recited in claim ~~40~~ 39, wherein the read/write policy comprises a rule to delete an earlier write request from the buffer when a subsequent write request to a same storage location on the non-volatile storage device is intercepted and buffered by the intermediate file system driver.

Claim 42. (currently amended) The system as recited in claim ~~38~~ 39, wherein an application executing on the processor registers with the intermediate file system driver to indicate compliance with selective buffering techniques to be used in conjunction with the read/write policy.